Pandas

In [1]:
```
import pandas as pd
```

In [2]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
#creating empty series
ser1=pd.Series()
```

```
C:\Users\qumrul hoda\AppData\Local\Temp\ipykernel_24096\856814744.py:2: FutureWarn
ing: The default dtype for empty Series will be 'object' instead of 'float64' in a
future version. Specify a dtype explicitly to silence this warning.
  ser1=pd.Series()
```

In [4]:
```
print(ser1)
```

```
Series([], dtype: float64)
```

In [6]:
```
#simple array
data=np.array(["A","B","1","c","d"])
```

In [7]:
```
ser2=pd.Series(data)
```

In [8]:
```
ser2
```

Out[8]:
```
0    A
1    B
2    1
3    c
4    d
dtype: object
```

In [15]:
```
df=pd.DataFrame()
print(df)

#list of string
list1={"Intro","to","pandas","lib"}

#calling Dataframe constructor on list
df=pd.DataFrame(list1)
```

```
Empty DataFrame
Columns: []
Index: []
```

In [16]:
```
data={"col1":[40,30,50],"col2":[34,20,44]}
df=pd.DataFrame(data)
```

In [17]:
```
df
```

Out[17]:

	col1	col2
0	40	34
1	30	20
2	50	44

In [18]:
```
df.shape
```

Out[18]:
```
(3, 2)
```

In [19]:
```
#we can creat mix datatypes
ser3=pd.Series([1,2,3,"Pandss",4.3,-3.4])
```

In [20]:
```
ser3
```

Out[20]:
```
0         1
1         2
2         3
3    Pandss
4       4.3
5      -3.4
dtype: object
```

In [21]:
```
index=["a","b","c","d","e","f"]
ser3=pd.Series([1,2,3,"Pandss",4.3,-3.4],index=index)
```

In [22]:
```
ser3
```

Out[22]:
```
a         1
b         2
c         3
d    Pandss
e       4.3
f      -3.4
dtype: object
```

In [23]:
```
ser3["a"]
```

Out[23]:
```
1
```

In [24]:
```
ser3["d"]
```

Out[24]:
```
'Pandss'
```

In [25]:
```
#creating series using dictionaries
#key:value
```

In [26]:
```
ser4=pd.Series({"Qumrul":100,"Ritik":99,"Babul":60})
```

In [27]:
```
ser4
```

Out[27]:
```
Qumrul    100
Ritik      99
Babul      60
dtype: int64
```

In [28]:
```
#performing logical operation
```

In [30]:
```
ser4[ser4>60]
```

Out[30]:
```
Qumrul    100
Ritik      99
dtype: int64
```

In [5]:
```
#creating sample DataFrame
import pandas as pd
```

In [8]:
```
col1=["Qumrul","Ritik","Babul","Rayyan","Aasif"]
col2=[90,80,40,60,70]
df=pd.DataFrame({"Name":col1,"Marks":col2})
```

In [9]:
```
df
```

Out[9]:

	Name	Marks
0	Qumrul	90
1	Ritik	80
2	Babul	40
3	Rayyan	60
4	Aasif	70

In [10]:
```
#get only first 3 rows
```

In [11]:
```
df.head(3)
```

Out[11]:

	Name	Marks
0	Qumrul	90
1	Ritik	80
2	Babul	40

In [12]:
```
#get rows from below
df.tail(3)
```

Out[12]:

	Name	Marks
2	Babul	40
3	Rayyan	60
4	Aasif	70

In [13]:
```
#printing all column
df.columns
```

Out[13]:
```
Index(['Name', 'Marks'], dtype='object')
```

In [14]:
```
#change the column name
```

In [15]:
```
df.columns=["stu_Names","Stu_Marks"]
```

In [16]:
```
df
```

Out[16]:

	stu_Names	Stu_Marks
0	Qumrul	90
1	Ritik	80
2	Babul	40
3	Rayyan	60
4	Aasif	70

In [17]:
```
#change the row names
```

In [19]:
```
df.index=["a","b","c","d","e"]
df
```

Out[19]:

	stu_Names	Stu_Marks
a	Qumrul	90
b	Ritik	80
c	Babul	40
d	Rayyan	60
e	Aasif	70

In [20]:
```
#how we can get only one column
```

In [22]:
```
df["Stu_Marks"]
```

Out[22]:
```
a    90
b    80
c    40
d    60
e    70
Name: Stu_Marks, dtype: int64
```

In [23]:
```
df["stu_Names"]
```

Out[23]:
```
a    Qumrul
b     Ritik
c     Babul
d    Rayyan
e     Aasif
Name: stu_Names, dtype: object
```

In [24]:
```
#create another new dataframe
```

In [11]:
```
col1=["Qumrul","Ritik","Babul","Rayyan","Aasif"]
col2=[90,80,40,60,70]
col3=["A","C","D","B","C"]
df=pd.DataFrame({"Name":col1,"Marks":col2,"Grade":col3})
df
```

Out[11]:

	Name	Marks	Grade
0	Qumrul	90	A
1	Ritik	80	C
2	Babul	40	D
3	Rayyan	60	B
4	Aasif	70	C

In [26]:
```
#access rows from limit index
#here ending index is exclusive
```

In [12]:
```
df.iloc[1:4]
```

Out[12]:

	Name	Marks	Grade
1	Ritik	80	C
2	Babul	40	D
3	Rayyan	60	B

In [28]:
```
df.iloc[0:2]
```

Out[28]:

	Name	Marks	Grade
0	Qumrul	90	A
1	Ritik	80	C

In [29]:
```
#access all rows from index 1 to 3 but only grade column index 2
```

In [30]:
```
df.iloc[1:3,2] #df[rows, column]
```

Out[30]:
```
1    C
2    D
Name: Grade, dtype: object
```

In [31]:
```
#acccessing all rows and all columns
df.iloc[:,:]
```

Out[31]:

	Name	Marks	Grade
0	Qumrul	90	A
1	Ritik	80	C
2	Babul	40	D
3	Rayyan	60	B
4	Aasif	70	C

In [32]:
```
#performing mathematical operation
```

In [33]:
```
df[df["Marks"]<90]
```

Out[33]:

	Name	Marks	Grade
1	Ritik	80	C
2	Babul	40	D
3	Rayyan	60	B
4	Aasif	70	C

In [34]:
```
#who got grade A
df[df["Grade"]=="A"]
```

Out[34]:

	Name	Marks	Grade
0	Qumrul	90	A

In [6]:
```
import pandas as pd
```

In [13]:
```
#access multiple
df[["Name","Marks"]]
```

Out[13]:

	Name	Marks
0	Qumrul	90
1	Ritik	80
2	Babul	40
3	Rayyan	60
4	Aasif	70

Pandas Concatenate

it will helps to concatenate [stack] two dataframes either vertically or horizontally

In [14]:
```
import pandas as pd
```

In [16]:
```
col1=["Qumrul","Ritik","Babul","Rayyan","Aasif"]
df1=pd.DataFrame({"Names":col1})
df1
```

Out[16]:

	Names
0	Qumrul
1	Ritik
2	Babul
3	Rayyan
4	Aasif

In [17]:
```
col2=[90,80,40,60,70]
df2=pd.DataFrame({"Marks:":col2})
```

```
df2
```

Out[17]:

	Marks:
0	90
1	80
2	40
3	60
4	70

In [18]:
```
df3=pd.concat([df1,df2],axis=1,ignore_index=True)
```

In [19]:
```
df3
```

Out[19]:

	0	1
0	Qumrul	90
1	Ritik	80
2	Babul	40
3	Rayyan	60
4	Aasif	70

In [23]:
```
df3=pd.concat([df1,df2],axis=1)
```

In [24]:
```
df3
```

Out[24]:

	Names	Marks:
0	Qumrul	90
1	Ritik	80
2	Babul	40
3	Rayyan	60
4	Aasif	70

In [25]:
```
#apply method
```

In [26]:
```
import pandas as pd
```

In [29]:
```
#create dictionary
data={"Product_ID":[1,2,3,4,5],"Category":["A","B","C","D","C"]
     ,"Available_stock":[2,4,8,5,9],"Price":[7100,29000,34999,4444,5644]}
```

In [31]:
```
#convert the dictionary into dataframe
df=pd.DataFrame(data)
```

In [32]:
```
df
```

Out[32]:

	Product_ID	Category	Available_stock	Price
0	1	A	2	7100
1	2	B	4	29000
2	3	C	8	34999
3	4	D	5	4444
4	5	C	9	5644

In [37]:
```
df=pd.DataFrame(data)
print("Original DataFrame:\n",df)
```

```
Original DataFrame:
    Product_ID Category  Available_stock  Price
0           1        A                2   7100
1           2        B                4  29000
2           3        C                8  34999
3           4        D                5   4444
4           5        C                9   5644
```

In [39]:
```
df.head()
```

Out[39]:

	Product_ID	Category	Available_stock	Price
0	1	A	2	7100
1	2	B	4	29000
2	3	C	8	34999
3	4	D	5	4444
4	5	C	9	5644

In [43]:
```
df["Priceperunit_in_thousand"]=df["Price"].apply(lambda x:x/1000)
```

In [44]:
```
df.head()
```

Out[44]:

	Product_ID	Category	Available_stock	Price	Priceperunit_in_thousand
0	1	A	2	7100	7.100
1	2	B	4	29000	29.000
2	3	C	8	34999	34.999
3	4	D	5	4444	4.444
4	5	C	9	5644	5.644

In [48]:
```
df["TotalValue"]=df.apply(lambda row:row["Available_stock"]*row["Price"],axis=1)
```

In [49]:
```
df.head()
```

Out[49]:

	Product_ID	Category	Available_stock	Price	Priceperunit_in_thousand	TotalValue
0	1	A	2	7100	7.100	14200
1	2	B	4	29000	29.000	116000
2	3	C	8	34999	34.999	279992
3	4	D	5	4444	4.444	22220
4	5	C	9	5644	5.644	50796

In [54]:
```
def func(x):
    return x/1000
```

In [55]:
```
df["Total_value_in_thousand"]=df["TotalValue"].apply(func)
```

In [56]:
```
df.head()
```

Out[56]:

	Product_ID	Category	Available_stock	Price	Priceperunit_in_thousand	TotalValue	Total_value_
0	1	A	2	7100	7.100	14200	
1	2	B	4	29000	29.000	116000	
2	3	C	8	34999	34.999	279992	
3	4	D	5	4444	4.444	22220	
4	5	C	9	5644	5.644	50796	

In [57]:
```
col1=["Qumrul","Ritik","Babul","Rayyan","Aasif"]
col2=[90,80,40,60,70]
col3=["A","C","D","B","C"]
df=pd.DataFrame({"Name":col1,"Marks":col2,"Grade":col3})
df
```

Out[57]:

	Name	Marks	Grade
0	Qumrul	90	A
1	Ritik	80	C
2	Babul	40	D
3	Rayyan	60	B
4	Aasif	70	C

In [58]:
```
df.loc[2]
```

Out[58]:
```
Name      Babul
Marks        40
Grade         D
Name: 2, dtype: object
```

In [59]:
```
df.iloc[2]
```

Out[59]:
```
Name      Babul
Marks        40
Grade         D
Name: 2, dtype: object
```

In [60]:
```
df.iloc[2][["Marks","Grade"]]
```

Out[60]:
```
Marks    40
Grade     D
Name: 2, dtype: object
```

pandas GroupBy

In [1]:
```
#pandas dataframe.groupby() function is used to split the into groups based on some
#Pandas groupby is used for grouping the data according to the categories and apply
```

In [2]:
```
import pandas as pd
```

In [3]:
```
col1=["Qumrul","Ritik","Babul","Rayyan","Aasif"]
col2=[90,80,40,60,70]
col3=["A","C","D","B","C"]
df=pd.DataFrame({"Name":col1,"Marks":col2,"Grade":col3})
df
```

Out[3]:

	Name	Marks	Grade
0	Qumrul	90	A
1	Ritik	80	C
2	Babul	40	D
3	Rayyan	60	B
4	Aasif	70	C

In [4]:
```
#let see who have scored top marks among all of them
```

In [5]:
```
df.groupby("Name").agg("sum")
```

Out[5]:

	Marks
Name	
Aasif	70
Babul	40
Qumrul	90
Rayyan	60
Ritik	80

In [ ]:
```

```